

Instalación MongoDB personalizada

Preparación del servidor

Es una buena práctica planificar y gestionar desde el principio la seguridad y permisos de las aplicaciones y usuarios que tengan acceso a nuestro servidor. Al igual que en una empresa no todos los empleados tienen acceso a todas las instalaciones, en un servidor pasa exactamente lo mismo. En general, es una mala práctica realizar todas las acciones con el usuario root.

A lo largo del curso usaremos los nodos para desempeñar uno o varios papeles y sería bueno que hubiera un usuario para cada uno de ellos. Ojo, no confundir el usuario del servidor con capacidad para gestionar el servicio con los usuarios que pueden usar el servicio.

```
administrador@ubuntuuserver:~$ sudo adduser mongodb
[sudo] password for administrador:
Adding user `mongodb' ...
Adding new group `mongodb' (1001) ...
Adding new user `mongodb' (1001) with group `mongodb' ...
Creating home directory `/home/mongodb' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mongodb
Enter the new value, or press ENTER for the default
  Full Name []:
   Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
administrador@ubuntuuserver:~$
```

```
sudo adduser mongodb
```

Con este comando creamos un usuario llamado mongodb en el equipo servidor. Como parte del proceso de creación, el sistema operativo nos preguntará la contraseña de este nuevo usuario y algunos datos que podemos dejar en blanco. También creará un grupo y la carpeta home con su mismo nombre. Debemos tener en cuenta que la contraseña que le pongamos a este usuario será la que se nos solicite cada vez que ingresemos con este usuario. También es importante entender que este será el usuario que tendrá permisos sobre los ejecutables de mongodb. Lanzar mongodb con permisos administrativos siempre funcionará pero no es una buena práctica

```
sudo usermod -sG sudo mongodb
```

Con este comando añadimos el usuario mongodb que acabamos de crear en el paso anterior al grupo sudo. Lo que estamos haciendo es dar la posibilidad al usuario mongodb de realizar alguna tarea con permisos administrativos y utilizar “sudo” cuando lo necesite.

Como parte de la preparación del servidor también es necesario decidir donde guardaremos los archivos ejecutables de mongo y donde guardaremos los datos de la base de datos. Esta decisión da para mucho debate. Alguien podría defender con argumentos que todo debería estar dentro del home del usuario mongodb. Otros pueden defender que todo debería estar fuera del home por ejemplo en la carpeta “/opt”. Allí se pueden configurar la seguridad con los permisos adecuados y a la vez estar disponibles para otros usuarios sin tener que pasar por el home de otro usuario.

En esta guía tomaremos un enfoque intermedio, dejaremos los ejecutables y el log dentro del home del usuario mongodb. Por otro lado, los datos quedarán fuera del home para que puedan estar disponibles a otros usuarios autorizados, concretamente en la ruta “/opt/datos”. Ya está decidido, volveremos a este punto en cuanto nos descarguemos los ejecutables de mongo desde su web oficial.

Instalación descargando las fuentes.

En este momento ya tenemos creado el usuario que tendrá permisos y responsabilidades para el manejo del servidor de mongodb por lo que cambiamos de usuario para que en todas las siguientes acciones, el propietario ya sea el usuario mongodb.

```
administrador@ubuntuuserver:~$ su mongodb
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mongodb@ubuntuuserver:/home/administrador$ cd /home/mongodb/
mongodb@ubuntuuserver:~$ pwd
/home/mongodb
mongodb@ubuntuuserver:~$ _
```

```
su mongodb
cd /home/mongodb
```

Con estos comandos cambiamos de usuario y nos situamos en su carpeta home.

Usaremos la versión MongoDB Community Edition. Desde la web <https://www.mongodb.com/download-center/community/releases> podemos seleccionar el archivo adecuado en función de la arquitectura de nuestra máquina y el sistema operativo instalado en ella.

```
mongodb@ubuntuuserver:~$ wget https://fastdl.mongodb.org/linux/mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
--2023-11-29 10:58:29-- https://fastdl.mongodb.org/linux/mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
Resolving fastdl.mongodb.org (fastdl.mongodb.org)... 13.224.115.58, 13.224.115.54, 13.224.115.61, ...
Connecting to fastdl.mongodb.org (fastdl.mongodb.org)[13.224.115.58]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 80190188 (76M) [application/gzip]
Saving to: 'mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz'

mongodb-linux-aarch64-ubuntu2204-7.0.3. 100%[=====] 76,47M 58,5MB/s in 1,3s

2023-11-29 10:58:30 (58,5 MB/s) - 'mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz' saved [80190188/80190188]

mongodb@ubuntuuserver:~$
```

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
```

Mediante el comando wget descargamos un archivo desde la ruta indicada. En mi caso he seleccionado el paquete para el sistema operativo Ubuntu 22.04 sobre arquitectura ARM64. Este caso es una combinación poco habitual así que tendremos que averiguar cuál es la ruta adecuada para caso. Si tenemos dudas deberíamos probar con la versión amd64 que es válida para procesadores Intel y amd.

```
mongodb@ubuntuuserver:~$ tar -zxvf mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
mongodb-linux-aarch64-ubuntu2204-7.0.3/LICENSE-Community.txt
mongodb-linux-aarch64-ubuntu2204-7.0.3/MPL-2
mongodb-linux-aarch64-ubuntu2204-7.0.3/README
mongodb-linux-aarch64-ubuntu2204-7.0.3/THIRD-PARTY-NOTICES
mongodb-linux-aarch64-ubuntu2204-7.0.3/bin/install_compass
mongodb-linux-aarch64-ubuntu2204-7.0.3/bin/mongodb
mongodb-linux-aarch64-ubuntu2204-7.0.3/bin/mongos
mongodb@ubuntuuserver:~$ ls -la
total 78336
drwxr-x--- 3 mongodb mongodb 4096 nov 29 11:04 .
drwxr-xr-x 4 root root 4096 nov 29 10:45 ..
-rw-r--r-- 1 mongodb mongodb 220 nov 29 10:45 .bash_logout
-rw-r--r-- 1 mongodb mongodb 3771 nov 29 10:45 .bashrc
drwxrwxr-x 3 mongodb mongodb 4096 nov 29 11:04 mongodb-linux-aarch64-ubuntu2204-7.0.3
-rw-rw-r-- 1 mongodb mongodb 80190188 oct 31 15:00 mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
-rw-r--r-- 1 mongodb mongodb 807 nov 29 10:45 .profile
mongodb@ubuntuuserver:~$
```

```
tar -zxvf mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
```

Una vez descargado el paquete debemos descomprimirlo con el comando anterior y aprovechar para mover la carpeta a la ubicación que queramos.

```
mv mongo-linux-aarch64-ubuntu2204-7.0.3 mongo
```

En mi caso, como acordamos anteriormente, dejaré los ejecutables de MongoDB en la carpeta mongo dentro de la carpeta personal del usuario mongodb.

```
mongodb@ubuntuuserver:~$ ls -la
total 78336
drwxr-x--- 3 mongodb mongodb 4096 nov 29 11:10 .
drwxr-xr-x 4 root root 4096 nov 29 10:45 ..
-rw-r--r-- 1 mongodb mongodb 220 nov 29 10:45 .bash_logout
-rw-r--r-- 1 mongodb mongodb 3771 nov 29 10:45 .bashrc
drwxrwxr-x 3 mongodb mongodb 4096 nov 29 11:04 mongo
-rw-rw-r-- 1 mongodb mongodb 80190188 oct 31 15:00 mongodb-linux-aarch64-ubuntu2204-7.0.3.tgz
-rw-r--r-- 1 mongodb mongodb 807 nov 29 10:45 .profile
mongodb@ubuntuuserver:~$
```

Configuración de mongodb

Si exploramos dentro de la carpeta mongo veremos que lo interesante está en los dos archivos ejecutables que están en la ruta `"/home/mongodb/mongo/bin"` Allí tenemos el archivo `"mongod"` encargado de la parte servidora y el archivo `"mongos"` encargado de la interfaz de comandos.

Para que los archivos binarios estén accesibles desde cualquier ruta del sistema operativo será necesario añadirlos al **PATH** o crear un enlace simbólico de esos archivos ejecutables en la carpeta `/usr/local/bin`. En este caso vamos a editar el archivo `"/home/mongodb/.bashrc"`.

```
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH=$PATH:/home/mongodb/mongo/bin
```

```
sudo nano /home/mongodb/.bashrc
```

Con este comando editamos el archivo `".bashrc"` del usuario mongodb donde se guardan las configuraciones de inicio de este usuario. Al final del archivo añadiremos una línea para indicar que al PATH actual se le añade una nueva ruta. Ya que este archivo se lee al arrancar el equipo y lo acabamos de modificar deberíamos cargarlo de nuevo explícitamente con el comando `source` o, más sencillo en este caso, reiniciando el equipo.

```
mongodb@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/mongodb/mongo/bin
mongodb@ubuntu:~$ _
```

```
echo $PATH
```

Con este comando listamos las rutas donde el sistema operativo buscará ejecutables. Debería aparecer la ruta de los ejecutables de mongo entre otros.

Si ahora ejecutamos el servidor mongod tendremos problemas porque la configuración mínima que mongo espera encontrar aún no existe.

```
mongodb@ubuntu:~$ mongod
{"t":{"sdate":"2023-11-29T14:07:31.551+00:00","s":"I","c":"NETWORK","id":4915701,"ctx":"main","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"isInternalClient":true}}},"t":{"sdate":"2023-11-29T14:07:31.551+00:00","s":"I","c":"CONTROL","id":23285,"ctx":"main","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"},"t":{"sdate":"2023-11-29T14:07:31.552+00:00","s":"I","c":"NETWORK","id":4648601,"ctx":"main","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."},"t":{"sdate":"2023-11-29T14:07:31.553+00:00","s":"I","c":"REPL","id":5123008,"ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"},"t":{"sdate":"2023-11-29T14:07:31.554+00:00","s":"I","c":"REPL","id":5123008,"ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"},"t":{"sdate":"2023-11-29T14:07:31.554+00:00","s":"I","c":"CONTROL","id":5945603,"ctx":"main","msg":"Multi threading initialized"},"t":{"sdate":"2023-11-29T14:07:31.554+00:00","s":"I","c":"TENANT_M","id":7091600,"ctx":"main","msg":"Starting TenantMigrationAccessBlockerRegistry"},"t":{"sdate":"2023-11-29T14:07:31.555+00:00","s":"I","c":"CONTROL","id":4615611,"ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":1035,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"ubuntu:~$"},"t":{"sdate":"2023-11-29T14:07:31.555+00:00","s":"I","c":"CONTROL","id":23403,"ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"7.0.3","gitVersion":"b96efb7e0cf6134d5938de8a94c37cec3f22cfff4","opensslVersion":"OpenSSL 3.0.2 15 Mar 2022","modules":[],"allocator":"tcmalloc","environment":{"distmod":"ubuntu2204","distarch":"aarch64","target_arch":"aarch64"}}},"t":{"sdate":"2023-11-29T14:07:31.555+00:00","s":"I","c":"CONTROL","id":51765,"ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Ubuntu","version":"22.04"}}},"t":{"sdate":"2023-11-29T14:07:31.555+00:00","s":"I","c":"CONTROL","id":21951,"ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"}}}","t":{"sdate":"2023-11-29T14:07:31.558+00:00","s":"E","c":"CONTROL","id":20557,"ctx":"initandlisten","msg":"DBException in initAndListen, terminating","attr":{"error":"NonExistentPath: Data directory /data/db not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the 'storage.dbPath' option in the configuration file."},"t":{"sdate":"2023-11-29T14:07:31.558+00:00","s":"I","c":"REPL","id":4784900,"ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown","attr":{"waitTimeMillis":15000},"t":{"sdate":"2023-11-29T14:07:31.559+00:00","s":"I","c":"REPL","id":4794602,"ctx":"initandlisten","msg":"Attempting to enter quiesce mode"},"t":{"sdate":"2023-11-29T14:07:31.560+00:00","s":"I","c":"REPL","id":6371601,"ctx":"initandlisten","msg":"Shutting down the FLE Crud thread pool"},"t":{"sdate":"2023-11-29T14:07:31.560+00:00","s":"I","c":"REPL","id":4784901,"ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"},"t":{"sdate":"2023-11-29T14:07:31.560+00:00","s":"I","c":"SHARDING","id":4784902,"ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"},"t":{"sdate":"2023-11-29T14:07:31.560+00:00","s":"I","c":"SHARDING","id":20562,"ctx":"initandlisten","msg":"Shutdown: going to close listening sockets"},"t":{"sdate":"2023-11-29T14:07:31.560+00:00","s":"I","c":"NETWORK","id":4784905,"ctx":"initandlisten","msg":"Shutting down the global connection pool"},"t":{"sdate":"2023-11-29T14:07:31.561+00:00","s":"I","c":"NETWORK","id":4784906,"ctx":"initandlisten","msg":"Shutting down the FlowControlTicketHolder"},"t":{"sdate":"2023-11-29T14:07:31.561+00:00","s":"I","c":"CONTROL","id":20520,"ctx":"initandlisten","msg":"Stopping further Flow Control ticket acquisitions"},"t":{"sdate":"2023-11-29T14:07:31.562+00:00","s":"I","c":"NETWORK","id":4784918,"ctx":"initandlisten","msg":"Shutting down the ReplicaSetMonitor"},"t":{"sdate":"2023-11-29T14:07:31.562+00:00","s":"I","c":"SHARDING","id":4784921,"ctx":"initandlisten","msg":"Shutting down the MigrationUtilExecutor"},"t":{"sdate":"2023-11-29T14:07:31.562+00:00","s":"I","c":"RSIO","id":22582,"ctx":"MigrationUtil-TaskExecutor","msg":"Killing all outstanding egress activity"},"t":{"sdate":"2023-11-29T14:07:31.563+00:00","s":"I","c":"COMMAND","id":4784923,"ctx":"initandlisten","msg":"Shutting down the ServiceEntryPoint"},"t":{"sdate":"2023-11-29T14:07:31.563+00:00","s":"I","c":"COMMAND","id":4784928,"ctx":"initandlisten","msg":"Shutting down the TTL monitor"},"t":{"sdate":"2023-11-29T14:07:31.563+00:00","s":"I","c":"CONTROL","id":6278511,"ctx":"initandlisten","msg":"Shutting down the Change Stream Expired Pre-Images Remover"},"t":{"sdate":"2023-11-29T14:07:31.564+00:00","s":"I","c":"CONTROL","id":4784929,"ctx":"initandlisten","msg":"Acquiring the global lock for shutdown"},"t":{"sdate":"2023-11-29T14:07:31.564+00:00","s":"I","c":"CONTROL","id":4784931,"ctx":"initandlisten","msg":"Dropping the scope cache for shutdown"},"t":{"sdate":"2023-11-29T14:07:31.564+00:00","s":"I","c":"CONTROL","id":20565,"ctx":"initandlisten","msg":"Now exiting"},"t":{"sdate":"2023-11-29T14:07:31.564+00:00","s":"I","c":"CONTROL","id":23138,"ctx":"initandlisten","msg":"Shutting down","attr":{"exitCode":100}}
```

```
mongod
```

En la captura anterior podemos ver como en el proceso de arranque del servidor mongo, intenta localizar la ruta de almacenamiento de datos por defecto que aún no existe y por tanto se para.

El proceso de instalación de mongo pasa por indicar dos rutas importantes. Una es la ruta donde mongo guardará el log y la otra es la ruta donde se guardarán los datos de la base de datos.

El log lo dejaremos dentro de la carpeta personal del usuario mongodb.

```
mkdir /home/mongodb/log
```

El siguiente paso en la configuración consiste en indicar la carpeta donde se guardarán los datos. Recordamos que anteriormente acordamos que la carpeta que alojaría los datos de la base de datos estaría en `“/opt/datos”`. La ruta de almacenamiento de los datos la dejaremos fuera de la carpeta personal del usuario `mongodb`. De esta manera podemos montar la ruta apuntando a un RAID, una carpeta remota o simplemente una carpeta más accesible para el resto de los usuarios.

```
sudo mkdir /opt/datos
sudo chown -R mongodb:mongodb /opt/datos
```

Con estos comandos creamos una carpeta llamada `datos` dentro de la carpeta del sistema `“/opt”`. Para escribir en esta carpeta necesitamos permisos administrativos. Además, el segundo comando sirve para que el usuario `mongodb` sea el propietario de esta carpeta y todo su contenido. Tendremos que hacerlo para que el usuario `mongodb`, que es el que va a lanzar el servidor, pueda escribir y leer en esa ruta sin necesidad de pedir permisos administrativos.

En este momento podemos confirmar que funciona lanzando el siguiente comando que, como podemos ver lleva explícita la ruta donde se guardan los datos.

```
mongod --dbpath=/opt/datos &
```

El símbolo de `“&”` al final del comando indica que queremos que lo ejecute en segundo plano. Si todo ha ido bien veremos que la última línea nos indica que está esperando por conexiones a través del puerto por defecto de `mongodb`, el 27017.

```

{ "t": { "sdate": "2023-11-29T14:22:53.443+00:00", "s": "I", "c": "STORAGE", "id": 5071100, "ctx": "initandlisten", "msg": "Clearing temp directory" },
{ "t": { "sdate": "2023-11-29T14:22:53.444+00:00", "s": "I", "c": "CONTROL", "id": 6608200, "ctx": "initandlisten", "msg": "Initializing cluster server parameters from disk" },
{ "t": { "sdate": "2023-11-29T14:22:53.445+00:00", "s": "I", "c": "CONTROL", "id": 20536, "ctx": "initandlisten", "msg": "Flow Control is enabled on this deployment" },
{ "t": { "sdate": "2023-11-29T14:22:53.446+00:00", "s": "I", "c": "FTDC", "id": 20625, "ctx": "initandlisten", "msg": "Initializing full-time diagnostic data capture" },
{ "t": { "sdate": "2023-11-29T14:22:53.455+00:00", "s": "I", "c": "REPL", "id": 6015317, "ctx": "initandlisten", "msg": "Setting new configuration state", "attr": { "newState": "ConfigReplicationDisabled", "oldState": "ConfigPreStart" } },
{ "t": { "sdate": "2023-11-29T14:22:53.464+00:00", "s": "I", "c": "STORAGE", "id": 22262, "ctx": "initandlisten", "msg": "Timestamp monitor starting" },
{ "t": { "sdate": "2023-11-29T14:22:53.466+00:00", "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "attr": { "address": "/tmp/mongodb-27017.sock" } },
{ "t": { "sdate": "2023-11-29T14:22:53.474+00:00", "s": "I", "c": "NETWORK", "id": 23015, "ctx": "listener", "msg": "Listening on", "attr": { "address": "127.0.0.1" } },
{ "t": { "sdate": "2023-11-29T14:22:53.479+00:00", "s": "I", "c": "NETWORK", "id": 23016, "ctx": "listener", "msg": "Waiting for connections", "attr": { "port": 27017, "ssl": "off" } }

```

Si revisamos los mensajes de salida veremos algunas líneas interesantes.

```

{ "t": { "sdate": "2023-11-26T16:39:50.586+00:00", "s": "W", "c": "CONTROL", "id": 22140, "ctx": "initandlisten", "msg": "This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning", "tags": [ "StartupWarning" ] }

```

Por ejemplo, en la línea de la captura anterior se puede ver como las conexiones están limitadas al localhost por defecto y, de querer que sea así, nos pide que se lo indiquemos por parámetro en el momento de lanzar el servidor.

Creando archivo de configuración.

En definitiva, podemos cargar el servidor mongo con una larga lista de parámetros para personalizar nuestra instancia de mongo o bien, una mejor práctica, podemos crear un archivo de configuración con esa misma información.

Podemos crear el archivo de configuración en cualquier ruta y con cualquier nombre pero en este caso usaremos la ruta y nombre oficial. Llamaremos a este archivo *mongod.conf* y lo guardaremos en la ruta */etc*.

```
GNU nano 6.2 /etc/mongod.conf *
# Consulta el resto de parámetros en:
# www.mongodb.com/docs/manual/reference/configuration-options

# Configuración de red. Direcciones que pueden conectarse y puerto de escucha
net:
  bindIp: 0.0.0.0
  port: 27017

# Ruta de almacenamiento de datos
storage:
  dbPath: /opt/datos

# Ruta de registro de logs
systemLog:
  destination: file
  path: /home/mongodb/log/mongo.log
  logAppend: true_
```

La captura anterior muestra lo que podía ser un punto de partida para el archivo de configuración */etc/mongod.conf*.

```
sudo nano /etc/mongod.conf
```

Con este comando creamos y editamos el archivo de configuración del servidor de mongodb. Si copiamos las líneas de la captura anterior configuraremos el servidor de mongo para que escuche peticiones de cualquier IP a través del puerto 27017. También le indicamos que la ruta que debe usar para almacén de datos es */opt/datos*. Debemos tener en cuenta que la ruta de datos oficial suele ser en */data/db* de manera genérica y en */var/lib/mongodb* en particular si el sistema operativo es Ubuntu. Por último, también configuramos la ruta del archivo log en */home/mongodb/log* donde se guardará información de todos los eventos importantes del servidor.

En este momento ya tenemos el servidor listo para ser lanzado.

```
mongod --config /etc/mongod.conf
```

Con este comando lanzamos el servidor mongodb con los parámetros guardados en el archivo de configuración que indicamos. Hay que tener presente que si lo lanzamos de esta manera la terminal, concretamente la sesión, quedará bloqueada hasta que no se pare el servidor, algo que no debería pasar. Una manera de conseguir mantener la

terminal usable es lanzar el comando con un ampersand (&) al final del comando, con ello estaremos indicando que queremos lanzar el servidor en segundo plano y mantener la terminal para ejecutar otros comandos.

```
administrador@ubuntuserver:~$ mongod --config /etc/mongod.conf &
[1] 1009
administrador@ubuntuserver:~$
```

Una vez lanzado el servidor, volveremos a tener control sobre la terminal y tendremos el identificador de proceso (PID) por si queremos matar el proceso y para así con el servidor mongo. Si no sabemos matar el proceso reiniciamos el equipo.

Mongo como servicio

Una opción mucho más elegante que lanzar a mano el servidor consiste en ejecutarlo como un servicio que se arranque automáticamente al iniciar el equipo. Lanzar una aplicación como un servicio nos permite pararlo, relanzarlo y consultar su estado de una manera muy sencilla.

```
GNU nano 6.2
[Unit]
Description=Servidor de bases de datos MongoDB como servicio
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=mongodb
Group=mongodb
WorkingDirectory=/home/mongodb/mongo/bin
Environment="OPTIONS=--config /etc/mongod.conf"
ExecStart=/home/mongodb/mongo/bin/mongod $OPTIONS

[Install]
WantedBy=multi-user.target
```

Para crear el servicio debemos crear un nuevo archivo con extensión “.service” en la carpeta “/etc/systemd/system/”.

```
sudo nano /etc/systemd/system/mongod.service
```

Con este comando creamos y editamos el archivo *mongod.service* que completaremos con el código de la captura anterior.

Este archivo tiene tres partes. En la primera parte además de una descripción indicamos que es necesario que la red ya esté levantada y funcionando para lanzar el servidor. También indicamos que la red es un requisito necesario.

En el apartado de servicio configuramos la ruta del ejecutable y los parámetros que le acompañan. En el último apartado indicamos que el servicio se debe lanzar cuando el sistema operativo esté listo para atender múltiples terminales, es decir, cuando el arranque del sistema operativo haya terminado correctamente.

```
sudo systemctl start mongod.service
```

Con este comando arrancamos el servicio de Mongo que tenemos configurado. Tenemos otras palabras reservadas que podemos usar con systemctl como start, stop, restart, mask, unmask, status, enable y disable.

```
sudo systemctl status mongod.service
```

Este comando muestra la información actual del servicio mongod.service.



```
Ubuntu Server 22.04 [Running]
mongodb@server:~$ sudo systemctl status mongod.service
● mongod.service - Servidor de bases de datos MongoDB como servicio
   Loaded: loaded (/etc/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-11-29 18:02:31 UTC; 40s ago
     Main PID: 1016 (mongod)
        Tasks: 32 (limit: 2220)
       Memory: 250.9M
          CPU: 2.128s
       CGroup: /system.slice/mongod.service
               └─1016 /home/mongodb/mongo/bin/mongod --config /etc/mongod.conf

nov 29 18:02:31 server systemd[1]: Started Servidor de bases de datos MongoDB como servicio.
mongodb@server:~$ _
```

En la captura anterior podemos ver la salida del comando y leyéndola podemos comprobar que el servicio se está ejecutando. Si nos fijamos en la línea que empieza por “loaded...” veremos que indica “Disabled”. Esto quiere decir que en cada arranque el servicio no se cargará automáticamente y tendremos que hacer un “start” para arrancarlo de nuevo.

```
sudo systemctl enable mongod.service
```

Este comando nos permitirá que este servicio arranque automáticamente con cada arranque del sistema operativo.

```
mongodb@server:~$ sudo systemctl status mongod.service
• mongod.service - Servidor de bases de datos MongoDB como servicio
   Loaded: loaded (/etc/systemd/system/mongod.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-11-29 18:06:10 UTC; 1min 11s ago
 Main PID: 1175 (mongod)
    Tasks: 32 (limit: 2220)
   Memory: 169.3M
      CPU: 1.808s
   CGroup: /system.slice/mongod.service
           └─1175 /home/mongodb/mongo/bin/mongod --config /etc/mongod.conf

nov 29 18:06:10 server systemd[1]: Started Servidor de bases de datos MongoDB como servicio.
mongodb@server:~$
```

En la captura anterior podemos ver como el servicio mongod sigue ejecutándose y además ahora confirmamos que en el próximo arranque se iniciará automáticamente porque en la línea que empieza por “Loaded” ahora indica “enabled”.

Los modificadores del comando systemctl, mask y unmask sirven para bloquear y desbloquear servicios. Los usaríamos cuando queremos que algún servicio ni arranque al inicio ni se pueda lanzar con un start.

En este momento, la configuración de un servidor sencillo de MongoDB está lista.

Conclusiones

Realizar una instalación a mano tiene la ventaja de que puedes personalizar todo desde un primer momento. Además, ayuda a entender dónde están los archivos de configuración y lo que contienen.

Uno de los aspectos más importantes de esta práctica es que hemos creado un usuario en el sistema operativo para que se encargue de la base de datos mongo. Ojo, no confundir con los usuarios de la base de datos. En este momento nos estamos refiriendo a qué usuario del sistema operativo puede arrancar el servidor, pararlo o, por ejemplo, cambiar el puerto de escucha o las ips que tienen acceso. La gestión de usuarios a nivel interno de la base de datos es algo relacionado con la configuración de esta a nivel interno.

Ya con este usuario nos hemos descargado los ejecutables desde la web oficial. Aquí es importante ser consciente de la arquitectura y sistema operativo de cada escenario. Hoy lo estamos montando en una máquina virtual, mañana podemos estar montándolo en una instancia de amazon AWS y será necesario saber cuál es la imagen adecuada.

La instalación es tan sencilla como descomprimir los archivos en la ruta que queramos atendiendo a algún criterio. En este caso me ha parecido interesante dejar los ejecutables y el log dentro de la carpeta del usuario con permisos para lanzar la base de datos. Los datos han quedado en una carpeta común a la que hemos dado la propiedad al usuario que maneja la base de datos para que pueda leer y escribir dentro sin problema.

Podemos poner toda la configuración por parámetro al lanzar el ejecutable, pero es buena práctica crear un archivo de configuración y dejarlo en la carpeta “/etc”.

En caso de reinicio de la máquina, el usuario encargado de mongodb tendría que volver a lanzar el ejecutable con referencia al archivo de configuración. Este proceso se puede automatizar creando un servicio.

Al crear un servicio no solo podemos lanzar un servidor como mongo en cada arranque, sino que también podemos hacer que no se lance hasta que la red está configurada, que se vuelva a lanzar si por alguna razón se para, consultar su estado o directamente pararlo sin tener que buscar su proceso.